

# Implementation of an Internet-controlled system under variable delays

Alexandre Seuret, M. Termens-Ballester, A. Toguyeni, S. El Khattabi and Jean-Pierre Richard

**Abstract**—This work deals with the control and the observation of a remote system using *Internet* as a communication line. The process consists in a Slave part S, with poor computing capacity, connected to a Master M through the Internet. The internal times of both M and S are synchronized thanks to a common GPS clock. The global system must ensure some speed performance whatever the delay variations. The main technical restriction is that the delays are less than some known bound, even if the framework we consider takes into account the perturbations generated by the Internet communication delays, by the packet losses and by the sampling phenomenon as well. In what concerns the design of the observer and controller, most of the theorems to be used here refer to proofs given in [14], and this will allow for focussing on the implementation aspects.

## I. INTRODUCTION

In a networked control situation, communication links unavoidably introduce delays that have to be taken into account in the design of control/observation loops. In the case of Internet, these delays are variable (jitter phenomenon), unknown and asymmetric. By *asymmetric*, we mean the transmission delays from Master to Slave  $h_1(t)$  (shortly, M-to-S), and S-to-M  $h_2(t)$  are not equal. The packets are not necessarily routed *via* the same way and, if  $R(t)$  denotes the round trip time (RTT), one may have  $h_1(t), h_2(t) \neq R(t)/2$ . The delay *variation* is problematic since it forbids one to apply the classical, predictor-based control laws (see [5], [8], [11]). Recently, nevertheless, [15] generalized predictor techniques to the case of variable, known delays. It was applied to a single-owner Ethernet network, which case, contrarily to Internet [9], allows for a dynamical model of the delay to be available. In the case of *unknown* variable delays, one may apply independent-of-delay conditions, but this turns out to be conservative, even if [1] obtained some success in the particular case of symmetric delays. For symmetric varying delays again, [6], [7], [10] considered  $h_1(t) = h_2(t) \leq h_m$  by introducing an input buffer making the delay become constant and equal to its maximum value  $h_m$ . However, it is obvious that maximizing the delay up to its largest value may decrease the speed performance of the global, remote system. Then, the speed performance has to be figured out within the design phase, which problem was not explicitly considered. To end

this list, note that [4] considered asymmetric delays, but only in the case of constant ones, *i.e.*  $h_1(t) = h_1 \neq h_2(t) = h_2$ .

The present work considers the general case of unknown, asymmetric and varying delays ( $h_1(t) \neq h_2(t)$ ), together with other perturbations such as packet losses and sampling effects. Since a buffer strategy is involved, the speed performance will be explicitly considered.

## II. ASSUMPTIONS AND PERTURBATIONS

The delays generated by Internet are time-varying, asymmetric, unknown. Thus, only few realistic hypotheses can be assumed:

(1) The delay variations satisfy  $\dot{h}_i(t) \leq 1$ , which means that all the packets are re-organized in their chronological emission order. In the sequel, UDP protocol will be used and the re-ordering will be ensured by specific threads in M and S.

(2) Delays are assumed to have known upper-bounds  $h_{im}$ , so that  $h_{i0} \leq h_i(t) \leq h_{im}$  holds. Of course, an Internet link does not guarantee such a bound by nature, and it must come from an additional switching strategy. If a network congestion occurs, the quality of service becomes too low for a stabilization to be possible (*i.e.*, delays become too large). In this case, the Master/Slave strategy is not adequate anymore and an autonomous control of S has to be conducted so not to leave it without control. This autonomous control corresponds to some deteriorated but stabilized Slave state. The switching decision can be based on the maximum admissible delay  $h_{im}$  for which our approach guarantees the existence of control/observation gains.

Besides, the discrete nature of the implementation induces data-sampling. Denoting  $t_k$  a  $k^{th}$  sampling instant (in S or in M as well), we assume we know a maximum sampling interval  $T_s$ , so that:

$$t_{k+1} - t_k \leq T_s \quad \forall k \geq 0. \quad (1)$$

The corresponding sampling effects represent a possible disturbance to the stabilization of the remote system and must be taken into account in the observer and controller design as well. Instead of turning into discrete-time, recurrent equations, [2] has considered such sampling effects as continuous-time phenomena with variable time delays. Indeed, the sample  $g(t_k)$  of a function  $g(t)$  at time  $t_k$  can be written as:  $g(t_k) = g(t - [t - t_k]) = g(t - \tau(t))$ , which notation replaces the sample-and-hold with an additional delay  $\tau(t) = t - t_k$ ,  $t \in [t_k, t_{k+1}[$ . By this way, an aperiodic sampling is modeled as unknown delay with the upper-bound  $T_s$  defined by (1). This model has several advantages:

This work was partially supported by the Council of "Region Nord - Pas de Calais", grant ARCir RoboCoop. Details on the RoboCoop project can be found at <http://syner.free.fr/>. The authors are with LAGIS, CNRS UMR 8146, Ecole Centrale de Lille, BP 48, 59651 Villeneuve d'Ascq Cedex, France. seuret.alexandre, amand.toguyeni, samir.elkhattabi, jean-pierre.richard@ec-lille.fr

(a) It allows one to consider non constant sampling intervals, this means, the case where there is no constant period  $T_s$  such that  $t_k = kT_s$ . This situation, coming from the task scheduling, is usual in real time architectures.

(b) It also includes the problem of *lost packets*, which are considered as a lengthening of the sampling interval. If the  $k^{th}$  packet is lost, the sample at time  $t_k$  is lost as well and the sampling interval becomes  $t - t_{k+1}$  instead of  $t - t_k$ . Then, the upper-bound  $T_s$  has to be replaced with  $T = T_s + qT_s$ , where  $q$  is the maximum number of consecutively packets it is possible to loose.

(c) It is compatible with the transmission delays, without introducing any additional complexity. For instance,  $\delta_i(t_k) = h_i(t_k) + t - t_k$  denote the two global delays ( $i = 1, 2$ ) resulting (at time  $t$ ) from communication, sampling and packet loss. Note the limit case  $\delta_i(t) = 1$  occurs.

To finish with, we need additional information on the two delays S-to-M and M-to-S. With this aim in mind, we equip both S and M with a GPS antenna, which allows their clocks to be synchronized. The control and measurements packets will be sent together with “time-stamps” that permit to reconstruct the *non-symmetric* delay information. By this way, both M-to-S  $h_1(t)$  and S-to-M  $h_2(t)$  delays are separately reconstructed by the system, and not only the RTT.

### III. CONTROL OBJECTIVE

The exchanged data correspond to the control (sent by M to S) and to the output of the remote process (sent by S to M). For energy saving reasons, S is supposed not to have a large computation capacity. The control and observation complexity has to be concentrated in M. Our purpose is to guarantee the robustness and speed performances of the global M/S system. The global system must ensure the closed-loop stability and a guaranteed speed rate whatever the delay variation. We require the exponential stability with the rate  $\alpha$  (or  $\alpha$ -stability). In other words, there must be a real  $K \geq 1$  such that the solution  $x(t; t_0, \phi)$  starting at any time  $t_0$  from any initial function  $\phi$  satisfies:  $\|x(t, t_0, \phi)\| \leq K\|\phi\|e^{-\alpha(t-t_0)}$ .

We want to stress that obtaining the  $\alpha$ -stability of the M/S system in the conditions depicted above is not that easy. The Master receives the information he needs for the control computation after it has crossed the communication zone. The GPS-based estimation of the transmission delay, joined to the observer, allows the Master to know what was the Slave state at the instant the information was sent to the Master. Similarly, the control computed by M will be applied some time after it is sent to S, and this dead-time is not known in advance.

For simplicity, the Slave is considered to be a linear system. The global  $\alpha$ -stability, robust w.r.t. the delay, will be proven by using the same Lyapunov-Krasovskii functionals as in [14] together with a separation principle. This will make possible to compute the controller and observer gains by using LMI optimization.

### IV. FEATURES OF THE REMOTE SYSTEM

The main features of the system are depicted on Figure 1.

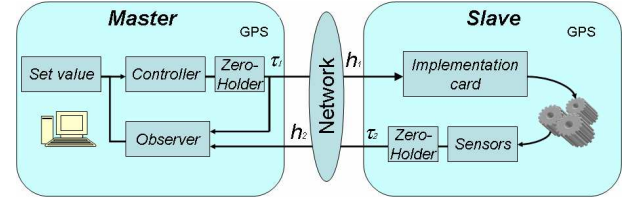


Fig. 1. Features of the remote system

(a) S can not build its own control. M computes and forwards the control to S. The forwarding suffers a delay  $\delta_1(t) = h_1(t_k) + \tau_1(t)$  induced by the communication, the sampling and the packet losses.

(b) S is driven by a linear, controllable and observable, known model  $S(A, B, C)$ , influenced by an input delay  $\delta_1^*(t) \geq \delta_1(t)$  to be defined later on (see “Receipt and processing the control data”):

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t - \delta_1^*(t)), \\ y(t) = Cx(t), \end{cases} \quad (2)$$

where  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ ,  $C \in \mathbb{R}^{p \times n}$ .

(c) S measures its output variables  $y(t)$  with a sampling delay  $\tau_2(t)$  (including the delay due to possible packet loss), which gives  $y_k = y(t - \tau_2(t))$ . M receives this sample after a delay  $h_2(t_k)$ .

(d) The sampling instants  $t_k$  may not be periodical (*i.e.*,  $t_k \neq kT_s$ ). It is supposed there is a known  $T_s$  such that (1) holds.

(e) M only can access  $y(t - \delta_2(t))$ , where  $\delta_2(t) = \tau_2(t) + h_2(t_k)$ . M includes an observer which aims at providing an estimation  $\hat{x}(t)$  of the complete Slave state  $x(t)$  at the present time  $t$ . From this estimation, the Master elaborates the control law.

(f) The two generated delays have known bounds  $\delta_{i0} = h_{i0}$  and  $\delta_{im} = h_{im} + T$ , so that  $\delta_{i0} < \delta_i(t) \leq \delta_{im}$  holds, and the delay variation satisfies  $\dot{\delta}_i(t) \leq 1$  (see Section II).

(g) Each part of the M/S system has a GPS card, which gives to M and S a shared clock. Each data packet includes a time-stamp (the time the packet was sent). By this way, the receiver can calculate the transfer delays  $h_i(t)$  as soon as it receives the packet.

#### The sampling delays

For any signal  $g(t)$ , we define the global delay  $\delta(t)$  which represents the combination of the sampling and packet loss delays  $\tau_k(t)$  (recall that  $\tau_k(t) \leq T = T_s + qT_s$ ) with the delay  $h(t_k)$  that the transmission line subjects to the packet containing the  $k^{th}$  sample:

$$\begin{aligned} g(t_k - h(t_k)) &= g(t - h(t_k) - (t - t_k)), \\ &= g(t - \delta(t)), \\ t_k \leq t < t_{k+1}, \quad \delta(t) &\triangleq h(t_k) + t - t_k. \end{aligned} \quad (3)$$

#### The control law

The controller in M computes a control law which takes into account some set value to be reached by S. The state feedback control  $u(t)$  is defined from the state estimate  $\hat{x}$  given by the observer, as follows:

$$u(t) = K\hat{x}(t). \quad (4)$$

The main difficulty is to determine the linear gain  $K$  of the state feedback control so to guarantee the stability of the Slave motion despite the value of the time-varying delay  $\delta_1(t)$ . This delay is not known by the Master when its control data is sent.

#### Transmission of the control data

The  $k^{th}$  data sent by M to S includes the control  $u(t_{1,k})$  it has just designed, together with the time  $t_{1,k}$  when the packet was sent. This packet goes across the Internet network. S receives this information at time  $t_{1,k}^r$ . Thanks to the GPS clock synchronization, this time has the same meaning for S as for M. Then the term  $t_{1,k}^r - t_{1,k}$ , corresponding to the transmission delay, is known by S once the packet has reached it.

#### Receipt and processing of the control data

The control, sent by M at time  $t_{1,k}$ , is received by S at time  $t_{1,k}^r > t_{1,k}$ . It will be injected in S input only at the pre-defined “target time”  $t_{1,k}^{target} = t_{1,k} + h_{1m}$ . This is achieved by storing the control during the additional waiting time  $t_{1,k} + h_{1m} - t_{1,k}^r$  (see Figure 2). This strategy is realistic because the delay is bounded by a known value (in general, one can choose  $h_{1m} = \delta_{1m}$ ). By this way, at any present time, M also knows the time  $t_{1,k}^{target}$  when this control  $u(t_{1,k})$  will be injected at the actuator input of S.

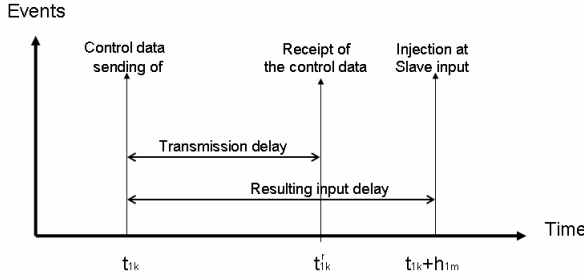


Fig. 2. Control data processing

#### Transmission of the measured output data

S accesses its output  $y$  at discrete instants. A sent packet contains the output  $y(t_{2,k'})$  together with its measurement instant  $t_{2,k'}$  which is the  $k'^{th}$  one. M receives at time  $t_{2,k'}^r$  the output data. Once the packet has reached M, the delay  $t_{2,k'}^r - t_{2,k'}$  is known thanks to the GPS synchronization.

#### Observation of the process

For a given  $k$  and for  $t \in [t_{1,k} + h_{1m}, t_{1,k+1} + h_{1m}]$ , there is a  $k'$  such that the proposed observer is of the form:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t_{1,k}) - L(y(t_{2,k'}) - \hat{y}(t_{2,k'})), \\ \hat{y}(t) = C\hat{x}(t). \end{cases} \quad (5)$$

The index  $k'$  corresponds to the most recent output information the Master M has received. Note that M knows the time  $t_{1,k}$  and the control  $u(t_{1,k})$  (see Section IV), which makes this observer realizable.

Using the delay re-writing proposed in (3), one obtains:

$$\begin{cases} \dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t - \delta_1(t)) \\ \quad - L(y(t - \delta_2(t)) - \hat{y}(t - \delta_2(t))), \\ \hat{y}(t) = C\hat{x}(t), \end{cases} \quad (6)$$

with  $\delta_1(t) \triangleq t - t_{1,k}$  and  $\delta_2(t) \triangleq t - t_{2,k'}$ .

In other words, the observer is realizable because the times  $t_{1,k}$  and  $t_{2,k'}$  defining the observer delays are known, thanks to the common GPS clock. The system features lead to  $\delta_1(t) \leq h_{1m} + T$  and  $\delta_2(t) \leq h_{2m} + T$ .

### V. DESIGN OF THE CONTROLLER AND OBSERVER GAINS

#### A. Exponential stabilization and notations

The controller and observer gains will have to be computed so to guarantee the optimal speed rate  $\alpha$  despite the presence of the various delays. The  $\alpha$ -stabilization results we shall use for the gain design refer to [14].

Because of the transmission, our two delays have a non zero lower bound (“non small delays” [3]), that we will generally write as follows:

$$\delta_i(t) = \delta_i + \eta_i(t), \quad \text{with } |\eta_i(t)| < \mu_i \quad \text{and} \quad \dot{\eta}_i(t) \leq 1. \quad (7)$$

This allows for using of a polytopic formulation of the variable delays, which was introduced in [12], [13]. This leads to the definition of the following extrema, that will be involved in our conditions for the gain design:

$$\begin{aligned} \beta_{11} &= e^{\alpha(\delta_1 - \mu_1)}, & \beta_{12} &= e^{\alpha(\delta_1 + \mu_1)}, \\ \beta_{21} &= e^{\alpha(\delta_2 - \mu_2)}, & \beta_{22} &= e^{\alpha(\delta_2 + \mu_2)}. \end{aligned} \quad (8)$$

#### B. Observer design

Since the pair  $(A, C)$  is observable, it is possible to determine a linear gain  $L$  such that the observer exponentially converges to the real system in the non-delayed case. The next theorem allows one to design another  $L$  so that the observer state  $\hat{x}(t)$  converges sufficiently fast (then, with exponential rate  $\alpha$ ) to the real system state  $x(t)$  despite a variable delay  $\delta_2(t)$  on the Slave output. The error vector is defined as  $e(t) = x(t) - \hat{x}(t)$ . From (2) and (6), it is ruled by:

$$\dot{e}(t) = Ae(t) - LCe(t - \delta_2(t)). \quad (9)$$

**Theorem 1:** Suppose that, for some positive scalars  $\alpha$  and  $\epsilon$ , there exists  $n \times n$  matrices  $0 < P_1, P, S, Y_1, Y_2, Z_1, Z_2, Z_3, R, R_a$  and a matrix  $W$  with appropriate dimensions such that the following LMI conditions are satisfied for  $j = 1, 2$ :

$$\begin{bmatrix} \Psi_2 & \begin{bmatrix} \beta_{2j}WC - Y_1 \\ \epsilon\beta_{2j}WC - Y_2 \end{bmatrix} & \mu_2\beta_{2j} \begin{bmatrix} WC \\ \epsilon WC \end{bmatrix} \\ * & -S & 0 \\ * & * & -\mu_2R_a \end{bmatrix} < 0, \quad (10)$$

$$\begin{bmatrix} R & Y \\ * & Z \end{bmatrix} \geq 0, \quad (11)$$

where  $\beta_{2j}$  are defined by (8) for  $j = 1, 2$  and the matrices  $Y, Z$  and  $\Psi_2$  are given by:

$$Y = [Y_1 \ Y_2], \quad Z = \begin{bmatrix} Z_1 & Z_2 \\ * & Z_3 \end{bmatrix}, \quad (12)$$

and:

$$\begin{aligned} \Psi_2^{11} &= P^T(A_0 + \alpha I) + (A_0 + \alpha I)^T P + S + \delta_2 Z_1 + Y_1 + Y_1^T, \\ \Psi_2^{12} &= P_1 - P + \epsilon P^T(A_0 + \alpha I)^T + \delta_2 \bar{Z}_2 + \bar{Y}_2, \\ \Psi_2^{22} &= -\epsilon(P + P^T) + \delta_2 \bar{Z}_3 + 2\mu_2 R_a, \end{aligned}$$

Then, the gain:

$$L = (P^T)^{-1}W, \quad (13)$$

makes the error (9) of observer (6) exponentially converge to the solution  $e(t) = 0$ , with the guaranteed decay rate  $\alpha > 0$ .

Note that the delay  $\delta_2(t)$  and then,  $\delta_2$  and  $\mu_2$ , are imposed by the various network perturbations. Using Theorem 1 means to tune  $\epsilon$  in order to maximize  $\alpha$ .

### C. Control design

First, we design of an ideal controller  $u = Kx$ , which means a perfect observer ( $e(t) = 0, x(t) = \hat{x}(t)$ ). Then, we consider:

$$\dot{x}(t) = Ax(t) + BKx(t - \delta_1(t)), \quad (14)$$

**Theorem 2:** Suppose that, for some positive numbers  $\alpha$  and  $\epsilon$ , there exists a positive definite matrix  $\bar{P}_1$ , matrices of size  $n \times n$ :  $\bar{P}, \bar{U}, \bar{Z}_1, \bar{Z}_2, \bar{Z}_3, \bar{Y}_1, \bar{Y}_2$  similarly to (12) and a  $n \times m$  matrix  $W$ , such that the following LMI conditions hold:

$$\begin{bmatrix} \Psi_3 & \begin{bmatrix} \beta_{1i}BW - \bar{Y}_1^T \\ \epsilon\beta_{1i}BW - \bar{Y}_2^T \end{bmatrix} & \mu_1 \begin{bmatrix} \beta_{1i}BW \\ \epsilon\beta_{1i}BW \end{bmatrix} \\ * & -\bar{S} & 0 \\ * & * & -\mu_1 \bar{R}_a \end{bmatrix} < 0, \quad \forall i = 1, 2, \quad (15)$$

$$\begin{bmatrix} \bar{R} & \bar{Y}_1 & \bar{Y}_2 \\ * & \bar{Z}_1 & \bar{Z}_2 \\ * & * & \bar{Z}_3 \end{bmatrix} \geq 0, \quad (16)$$

where  $\beta_{1i}$ , for  $i = 1, 2$ , are defined by (8) and

$$\begin{aligned} \bar{\Psi}_3^{11} &= (A_0 + \alpha I)\bar{P} + \bar{P}^T(A_0 + \alpha I)^T + \bar{S} \\ &\quad + \delta_1 \bar{Z}_1 + \bar{Y}_1 + \bar{Y}_1^T, \\ \bar{\Psi}_3^{12} &= \bar{P}_1 - \bar{P} + \epsilon \bar{P}^T(A_0 + \alpha I)^T + \delta_1 \bar{Z}_2 + \bar{Y}_2, \\ \bar{\Psi}_3^{22} &= -\epsilon(\bar{P} + \bar{P}^T) + \delta_1 \bar{Z}_3 + 2\mu_1 \bar{R}_a. \end{aligned}$$

Then, the gain:

$$K = W\bar{P}^{-1}, \quad (17)$$

exponentially stabilizes the system (14) with the decay rate  $\alpha$  for all delay  $\delta_1(t)$  satisfying (7).

As for the observer, using Theorem 2 means to tune  $\epsilon$  in order to maximize  $\alpha$ .

### D. The global remote system

The gains  $K$  and  $L$  have to be computed in such a way they exponentially stabilize the global Master-Slave-Observer system despite the variable delays  $\delta_1(t)$  and  $\delta_2(t)$ . This global system is:

$$\begin{cases} \dot{x}(t) = Ax(t) + BK\hat{x}(t - \delta_1(t)), \\ \dot{e}(t) = Ae(t) - LCe(t - \delta_2(t)), \end{cases} \quad (18)$$

which leads to:

$$\begin{cases} \dot{x}(t) = Ax(t) + BKx(t - \delta_1(t)) - BK e(t - \delta_1(t)), \\ \dot{e}(t) = Ae(t) - LCe(t - \delta_2(t)). \end{cases} \quad (19)$$

The exponential stability of the global system can be shown by using a separation principle: Indeed, (19) is under a block-triangular form, which diagonal subsystems have been proven to be stable in Theorems 1 and 2. Then the whole system is  $\alpha$ -stable with the smallest  $\alpha$  computed from Theorems 1 and 2.

## VI. DATA-PROCESSING STRUCTURE

This section shows the next step to the implementation and focusses on the data-processing structure of the remote system.

Communication through sockets is designed around the so-called Client/Server architecture. Two processes connected by network links are not treated identically but rather one is considered the server and the other the client. On the other side, the two entities have synchronized clocks because of the GPS utilization. They have access to the same time value at each step of the observation or control data computation.

### A. Structure of the Master

M is composed of four threads. These threads are programs which work independently. They are useful here because they permit to do parallel tasks in the same program. The structure is described in Figure 3.

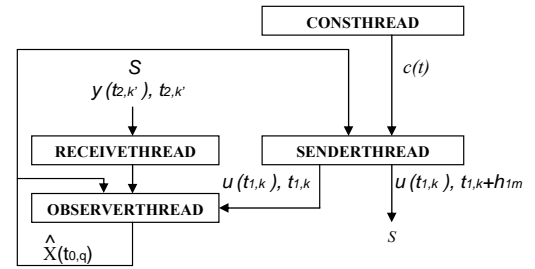


Fig. 3. Master Structure

(a) **OBSERVERTHREAD** concerns the computation of the estimate  $\hat{x}$  at the current time  $t_{0,q}$ . It works with a period  $T_0 \ll T_s$ , where  $T_s$  is defined in (1). The sampling instant  $t_{0,q}$  corresponds to the sampling of the **OBSERVERTHREAD**. It uses the last delayed output  $y(t_{2,k'})$  and the corresponding sampling instant  $t_{2,k'}$  delivered by S, as well as the control  $u(t_{1,k})$ . These data are retrieved from lists which contains a sequence of the past values of  $\hat{x}$  and  $u$ . Then, the thread computes the current estimate  $\hat{x}(t_{0,q+1})$  using (20) and adds it to the list.  $T_0$  must be small enough to obtain a computation instant nearly equal to  $t_{2,k'}$ . The estimate is computed as follows for  $i = 1, \dots, n$  and  $r = 1, \dots, p$ :

$$\begin{aligned} \hat{x}_i(t_{0,q+1}) &= \hat{x}_i(t_{0,q}) + \left[ \sum_{j=1}^n A_{(i,j)} \hat{x}_j(t_{0,q}) \right. \\ &\quad + \sum_{s=1}^m B_{(i,s)} u_s(t_{1,k} + h_{1m}) \\ &\quad \left. - \sum_{r=1}^p L_{(i,r)} (y_r(t_{2,k'}) - \hat{y}_r(t_{2,k'})) \right] T_0, \\ \hat{y}_r(0, t_{0,k}) &= \sum_{i=1}^n C_i \hat{x}(t_{0,q}) \end{aligned} \quad (20)$$

(b) **CONSTTHREAD** is a short program. Its purpose is to update the task value. The task values are updated periodically with a large period which allows to be driven the global system.

(c) **SENDERTHREAD** works periodically with a period  $T_0 \ll T_1 \leq T_s$ . S does not need a large number of control values. Its tasks are to receive the last estimate  $\hat{x}(t_{0,k+1})$  from the observer thread. Then it calculates the control  $u(t_{1,k+1}) = K\hat{x}(t_{1,k+1})$ . It regroups in a data-packet the control  $u(t_{0,k+1})$  and the sampling instant  $t_{0,k+1}$  in a list which is used by

OBSERVERTHREAD. Then at the sampling instant  $t_{1,k+1}$ , it sends the most recent data to the S.

(d) RECEIVERTHREAD is factual. This means it is only solicited when a data-packet arrives. It transfers the Slave information to the OBSERVERTHREAD.

### B. Structure of the Slave

As it was mentioned in section IV, S has a reduced computation power. So its tasks only consist in receiving and transmitting data from and to M. Figure 4 presents the data-processing structure of S.

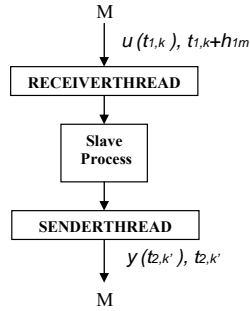


Fig. 4. Slave Structure

It consists in two threads:

(d) RECEIVERTHREAD is factual. This means it is only solicited when a data-packet arrives. It stores the control data  $u(t_{1,k})$  until the associated target time  $t_{1,k} + h_{1m}$  and, then, injects it at the slave input until the next time  $t_{1,k+1} + h_{1m}$ .  
 (b) SENDERTHREAD receives the last output information from the sensors,  $y(t_{2,k'})$  and notes the instant  $t_{2,k'}$ . Then it sends this data to M. This thread works periodically with a period  $T_2 \leq T_s$ .  $T_2$  has to be large enough to avoid a network congestion, which would increase  $h_{2m}$ , and small enough to increase the observer performance. However, note that any  $0 < T_2 \leq T_s$  can be chosen provided that the observer gain is computed from (13).

### VII. APPLICATION TO A MOBILE ROBOT

This study is illustrated on the model of a mobile robot (Slave S) which can move in one direction. The identification phase gives the following dynamics:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -11,32 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 11,32 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix}. \quad (21)$$

The characteristics of transmission delays combined with the sampling and computation effects lead to the values (see (7))  $\delta_1 = \delta_2 = 0.37s$ , and  $\mu_1 = \mu_2 = 0.11s$ . This values correspond to the sampling periods  $T_0 = 5ms$ ,  $T_1 = 100ms$  and  $T_2 = 100ms$  and to transmission delays  $h_1$  and  $h_2$  which lay in  $[0.16, 0.38]$  (thus,  $T_s = 100ms$ ). Theorem 1 applied to (9) guarantees that the error dynamics converge exponentially with  $\alpha = 1.01$  (obtained for  $\epsilon = 3.00$ ) if the gain  $L$  is chosen as:

$$L = \begin{bmatrix} -0.9119 \\ -0.0726 \end{bmatrix}. \quad (22)$$

Theorem 2 applied to (14) ensures the control law will exponentially stabilize the reduced system with  $\alpha = 1.01$ , obtained for  $\epsilon = 3.43$  and:

$$K = \begin{bmatrix} -0.9125 & -0.0801 \end{bmatrix}. \quad (23)$$

With these values, the global stability of the remote system (18) is also exponentially stable with a decay rate  $\alpha = 1.01$ .

Figure 5 shows a simulation result that was obtained for some particular delay variation law. On Figure 5a and 5b, the continuous model of the observer  $\hat{x}$  corresponds to the continuous curves, while the sampling instants correspond to the dashed lines. The output is driven to its set value. Figure 6 represents the sampled control send to S. Note that the following results were obtained using Matlab/Simulink. The concrete development is still in progress and will be done for the final version of this article.

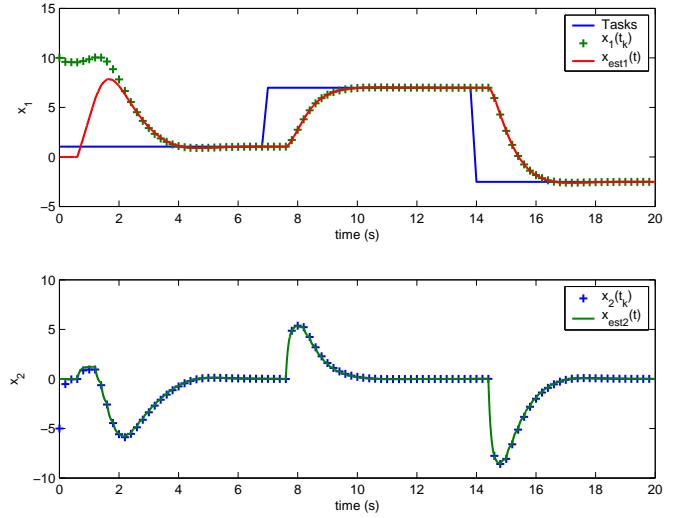


Fig. 5. Simulation results

### VIII. ACKNOWLEDGEMENTS

This study was realized within the RoboCoop project. The authors would like to thank Council of "Rgion Nord - Pas de Calais" (grant ARCir RoboCoop) for partially funding the project. Details on the RoboCoop project can be found at <http://syner.ec-lille.fr>.

### IX. CONCLUDING REMARKS

To finish with, let us recall the two main ideas in this control strategy: (1) the Master observer works in continuous time, whereas the Slave measurement is in discrete time. By this way, the observer reconstructs intermediate values of the Slave state. (2) All the perturbations generated by the network (delays, sampling, packet losses) are modeled in the form of time delays. Thanks to recent results on variable and non-small delays (*i.e.* delays which lower bound is non zero), the robust performance can be achieve with reduced assumptions.

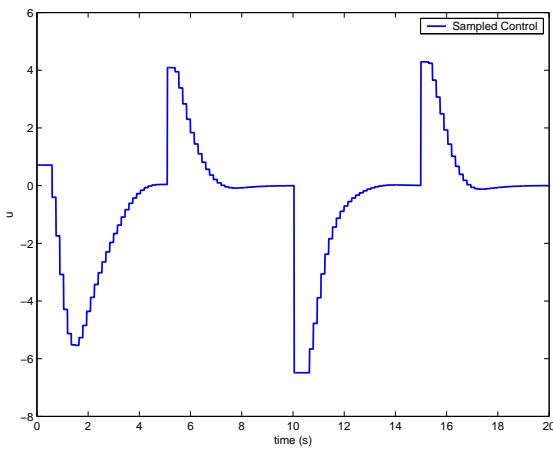


Fig. 6. The corresponding delayed and sampled control

## REFERENCES

- [1] A. Eusebi and C. Melchiorri, "Force-Reflecting telemanipulators with Time-delay: Stability Analysis and control design", IEEE trans. on Robotics and Automation, vol. 14(4), pp. 635-640, 1998.
- [2] E. Fridman A. Seuret, E. Fridman and J.-P. Richard, "Robust sampled-data stabilization of linear systems: An input delay approach", Automatica, vol. 40(8), pp. 1441-1446, 2004.
- [3] E. Fridman and U. Shaked, "A new  $H_\infty$  filter design for linear timedelay systems", IEEE Trans. Signal Processing, vol. 49, pp. 2839-2843, 2001.
- [4] C.E. Garcia, B. Morales, R. Carelli and J. Postigo, "Stability analysis for a teleoperation system with time delay and force feedback", Proceedings of the 39<sup>th</sup> Conference on Decision and Control Sydney, Australia December, 2000.
- [5] K. Gu, V.L. Kharitonov and J. Chen, "Stability of Time-Delay Systems", Birkhuser, Boston, 2003.
- [6] A. Leleve, P. Fraisse, P. Dauchez, F. Pierrot, "Modeling and simulation of robotic tasks teleoperated through the Internet Advanced Intelligent Mechatronics", Proceedings. IEEE/ASME International Conference pp.299 - 304 , 1999.
- [7] A. Leleve, P. Fraisse, P. Dauchez, "Telerobotics over IP networks: Towards a low-level real-time architecture", Proceedings. 2001 IEEE/RSJ International Conference on Volume 2, pp.643 - 648, 29 Oct.-3 Nov. 2001.
- [8] S.I. Niculescu, "Delay Effects on Stability", LNCIS No.269, Springer, 2001.
- [9] S.I. Niculescu, C.T. Abdallah and P.F. Hokayem, "Effects of channel dynamics on the stability of teleoperation", TDS04, 4<sup>th</sup> IFAC Workshop on Time Delay Systems, Rocquencourt, France, September 2003.
- [10] N. J. Ploplys, P. A. Kawka and A. G. Alleyne, "Closed-loop control over wireless networks", IEEE Control Systems Magazine, vol. 24(3), pp.58-71, june 2004.
- [11] J.-P. Richard, "Time delay systems: an overview of some recent advances and open problems", Automatica, vol. 39, pp. 1667-1694, 2004.
- [12] A. Seuret, M. Dambrine and J.-P. Richard, "Robust exponential stabilization for systems with time-varying delays", TDS04, 5<sup>th</sup> IFAC Workshop on Time Delay Systems, Leuven, Belgium, September 2004.
- [13] A. Seuret, E. Fridman and J.-P. Richard, "Sampled-data exponential stabilization of neutral systems with input and state delays", IEEE MED 2005, 13<sup>th</sup> Mediterranean Conference on Control and Automation, Cyprus, 2005.
- [14] A. Seuret, F. Michaut, J.-P. Richard and T. Divoux, "Networked Control using GPS Synchronization", ACC06, American Control Conf., Minneapolis, USA, June 2006.
- [15] E. Witrant, C. Canudas-de-Wit and D. Georges, "Remote Output Stabilization Under Two Channels Time-Varying Delays", 4<sup>th</sup> Workshop on Time Delay Systems, Rocquencourt, 2003.